

## SYSTEM AND METHOD FOR SINGLE SECURITY ADMINISTRATION

Inventors: Hong-Hsi Lo  
Rich Lee

5

### COPYRIGHT NOTICE

10

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

15

### Claim of Priority:

[0001] This application claims the benefit of U.S. Provisional Application "SYSTEM AND METHOD FOR SINGLE SECURITY ADMINISTRATION", Serial No. 60/432,125; filed December 9, 2002, and incorporated herein by reference.

20

### Field of the Invention:

[0002] The invention is generally related to application servers and other enterprise servers, and particularly to a system and method for administering security in complex or distributed server environments.

25

### Background:

[0003] In the application server marketplace, an enterprise-level customer will often utilize two or more different types of server product, some of which may be a newer version of a particular application server product, while others may be older versions, or even legacy systems. Many of these customers need the ability to implement an enterprise security framework that encompasses all of the

30

application servers. For example, a customer who uses both Tuxedo and WebLogic Server (WLS) might want to simplify their system administration work at the security level so that they can manage enterprise security from a central point. Today these customers will have two sets of security to manage - one set  
5 for their Tuxedo product, and the other set for their WLS product. If a particular user needs to access services in both the Tuxedo and the WLS environments then the system administrator will have to add or modify the same user information to both Tuxedo and WLS for that particular user. This repetition of administrative work is both cumbersome and error-prone. To date, there is no  
10 feature that provides the administrator with a centralized means for managing security that spans both (for example the Tuxedo and WebLogic) environments.

**Summary:**

**[0004]** The present invention solves the problem of managing security  
15 over different computing environments by consolidating all user related information at a central point, for example within an application server. With traditional security architectures, an enterprise system administrator who had both application servers (for example WebLogic Server, WLS) and other enterprise systems (for example Tuxedo) deployed within their environment,  
20 would usually have to manage two sets of security information, in this instance one for WebLogic Server and the other one for Tuxedo. The present invention leverages the application server's security to help the system administrator managing their security database, by eliminating user and group information from the enterprise system. System-specific information, such as Access Control List  
25 information can still reside in the Tuxedo product.

**[0005]** In accordance with one embodiment, a method is provided for providing single security administration comprising the steps of: allowing a client

(for example a Tuxedo client) to access a default security plugin; issuing a call (tpinit) to an Lightweight Directory Access Protocol (LDAP) authentication server at a first (e.g. Tuxedo) server; passing query user information from the LDAP authentication server to an embedded LDAP server at a second (e.g. WLS) server; returning corresponding user information to the LDAP authentication server; and, providing an authentication token for use by the client.

**Brief Description of the Drawings:**

**[0006]**        **Figure 1** illustrates a schematic of a single security system in accordance with an embodiment of the invention.

**[0007]**        **Figure 2** illustrates a flowchart of a method for providing single security administration in accordance with an embodiment of the invention.

**Detailed Description:**

**[0008]**        The present invention solves the problem of managing security over different computing environments by consolidating all user related information at a central point, for example within an application server. With traditional security architectures, an enterprise system administrator who had both application servers (for example the WebLogic Server product from BEA Systems, Inc., referred to herein as WLS), and other enterprise systems (for example the Tuxedo product, also from BEA Systems, Inc.), deployed within their environment, would usually have to manage two sets of security information, in this instance one for WebLogic Server and the other one for Tuxedo. The present invention leverages the application server's security to help the system administrator managing their security database, by eliminating user and group information from the enterprise system. System-specific information, such as

Access Control List information can still reside in the Tuxedo product.

**[0009]** More particularly, in an environment that includes either multiple application servers, or an application server and a legacy-type system, the present invention provides the system administrator with a means to manage their security database from a central point. In accordance with one embodiment, the application server's security features are leveraged to provide user authentication throughout the enterprise, which allows user and group information to be eliminated from the enterprise system. This new feature leverages OPEN Lightweight Directory Access Protocol (LDAP) to make a single user security data store and administration possible. The use of a single data store assists the system administrator by only requiring them to maintain user security information at a single location, for example at a WebLogic Server embedded LDAP server. The single security administration also means the system administrator can administer the security information from a single system, e.g. from within the WebLogic Server Console program.

**[0010]** In accordance with one embodiment, the system can be used to consolidate all user related information in WebLogic Server (WLS). Without this feature the system administrator of a Tuxedo/WLS environment would have to administer the user information separately in both Tuxedo and WLS. The administrator should be aware that the Tuxedo-specific tpgrp, and tpacl file information should still be maintained in Tuxedo if the Tuxedo ACL or MANDATORY\_ACL is desired.

#### DEFINITIONS OF TERMS, ACRONYMS, AND ABBREVIATIONS

**AAA:** Authentication, Authorization, and Auditing.

**ACL:** Access control list - The authorization scheme used by Tuxedo.

**LDAP:** Lightweight Directory Access Protocol - A standard way of managing directory information.

**MP:** The MP specifies a multi-machine configuration for a Tuxedo application.

**PIF:** Plug-In Framework - A Tuxedo infrastructure component that allows customization of BEA Tuxedo infrastructure capabilities through the use of plug-in module.

**Schema:** The Schema is used to define the structure of LDAP database. Every LDAP server must use a particular schema, which defines what attributes can be stored in what type of object.

**SDS:** System Data Store - The LDAP used internally by WLS default security service.

**UBBCONFIG:** The Tuxedo System /T ASCII configuration file.

**DN:** Distinguished Name.

**[0011]** **Figure 1** shows an illustration of a single security system 10 in accordance with an embodiment of the invention. In the past, the Tuxedo user security file would have to be copied from Tuxedo to each WLS server for use by that server in authenticating users. However, using the invention a Tuxedo customer can access the WLS security. As shown in **Figure 1**, a first enterprise or application server **12** (e.g. a Tuxedo server) communicates with a second enterprise or application server **14** (e.g. a WLS server). The first (Tuxedo) server provides an LDAP authentication server **16**. In this embodiment the LDAP authentication server replaces the regular Tuxedo authentication server, but from the perspective of a Tuxedo user operates much the same in that it continues to understand Tuxedo tpinit calls, etc. The second (WLS) server **14** includes an embedded LDAP server plugin **18** to allow the Tuxedo user to use WLS security.

When, for example, a Tuxedo client **20** makes a tpinit call, the Tuxedo library directs this call to the default security plugin **22**, which in turn forwards it **24** to the LDAP authentication server **16**. The LDAP authentication server **16** checks a user profile database (or user profile configuration information) to determine where the particular user security information is stored. An LDAP session is then initiated between the first (Tuxedo) server **12** and the second/determined (WLS) server **14**. A query user information **26** is passed from the LDAP authentication server **16** to the embedded LDAP server **18** at the WLS, specifying a particular user. The corresponding user information **28** is then returned to the LDAP authentication server **16**. Upon receipt of the user information a token **30** is then created that reflects this authentication result, and which can be used by the Tuxedo client.

**[0012]** The Tuxedo system shown in **Figure 1** does not need to be drastically changed in order to provide the distributed authentication - the LDAP authentication server is just configured so that it understands a standard Tuxedo tpinit call. From the clients' perspective the authentication process is the same as before (i.e. without single security implemented).

**[0013]** It should be noted that although a Tuxedo server and a WLS server is shown in **Figure 1**, the present system and methods can be implemented to work with other application servers and enterprise servers that support LDAP, or that support the use of an embedded LDAP authentication server. Furthermore, a cluster or plurality of servers can be used to implement single security administration, and to provide backup or failover authentication should one of the servers, or the communications link to one of the servers, fail. The failover server needs to be able to provide the LDAP authentication service.

**[0014]** In addition, in some embodiments a user information cache may

be provided in order to temporarily cache a copy of the user authentication information in case of a failure in the communications link between the two servers. Whenever necessary, the cache can be flushed to effectuate immediate changes to the security information.

5     **[0015]**       The system may also be fully scalable so that multiple authentication servers can be used, as can multiple embedded LDAP servers.

10    **[0016]**       **Figure 2** illustrates a flowchart of a method in accordance with an embodiment of the invention. As shown therein, in step 30, the client (in this instance a Tuxedo client) accesses the default security plugin. In step 32, the default security plugin issues a call (such as a tpinit call) to the LDAP authentication server. In step 34, the query user information is passed to the embedded LDAP server at the application server (for example the WebLogic Server). In step 36, corresponding user information is then returned to the LDAP authentication server. In step 38, the authentication server provides an authentication token for use by the client.

#### Migration Tool

20    **[0017]**       In addition to providing real-time authentication of users between different application server types, another application of the present system is that it can be used to perform migration of users from one system to another. For example, Tuxedo user information (from the Tuxedo tpusr and tpgrp files) can be migrated to WebLogic Server. During migration, a migrating utility, tpmigldap, takes input from the tpusr and tpgrp files and updates the corresponding WebLogic server security database.

25

#### Tuxedo/WLS Implementation

**[0018]** In the context of a Tuxedo / WebLogic server environment, an embodiment of the present invention allows Tuxedo to use WLS as the security database to authenticate Tuxedo users. In accordance with this embodiment, the single security database resides in WebLogic Server. It is required to modify the Tuxedo UBB configuration file to enable this new feature. At the same time Tuxedo can continue to support the old Tuxedo security authentication styles, such as NONE, APP\_PW, USER\_AUTH, ACL, and MANDATORY\_ACL. For customers with both Tuxedo and WebLogic Server, but who do not need to do tight integrating, they can continue configuring and operating Tuxedo as they had done before, i.e. have separate security database and separate security administration. However, the feature is especially useful for those customers who prefer to not maintain separate user databases.

**[0019]** The following is the list of the functions provided by this feature:

- Single User Security Database
- Single User Security Administration

#### User Characteristics and Impact

**[0020]** There is no visible impact to the Tuxedo user or to the server/client programming interface, since all of the changes to provide single security are “under the cover”. The default Tuxedo security type is "NONE" in the RESOURCES section of the UBBCONFIG file just as before. It can continue supporting other types of Tuxedo security, such as APP\_PW, USER\_AUTH, ACL, and MANDATORY\_ACL. The difference “under the cover” is that user authentication will actually retrieve user information from WLS instead of reading it from the tpusr file.

### Single User Security Database

**[0021]** In the current implementation, Tuxedo stores its security configuration information in three files. The tpusr file contains the Tuxedo user information including the password. The tpgrp file contains the Tuxedo group information. The tpacl file contains the Tuxedo Access Control List information. The enhanced functionality provided by the present invention allows Tuxedo to access the user security information stored in the WLS embedded LDAP server. The security information stored in WebLogic Server contains information such as user identification, password, and which security groups the user is a member. If this feature is used then there is no more need for tpusr file. However, the tpacl, and tpgrp files are still required if Tuxedo security is configured to use either ACL or MANDATORY\_ACL.

**[0022]** In one embodiment, for Tuxedo user information, the WLS LDAP database contains both the user name and password. The Tuxedo user name is mapped to the WLS System Data Store object class inetOrgPerson's "uid" that is the identity of login id. This is then set to the value entered for the user in the WLS Admin console. The user password is mapped to the "userpassword" in the inetOrgPerson class. The group which the user belongs to is mapped to the "wlsMemberOf" in the inetOrgPerson class.

**[0023]** For Tuxedo, group information is stored in the WLS groupOfURLs class. The name of the Tuxedo group is the common name (cn) attribute of the groupOfURLs. This group common name is set to the value entered for the group from the WLS Admin console. Table 1 describes these mappings.

	Tuxedo Name	LDAP class	LDAP attribute
	User name	InetOrgPerson	uid
5	User password	InetOrgPerson	userpassword
	User group information	InetOrgPerson	wlsMemberOf
	Group	GroupOfURLs	cn

Table 1

10

**[0024]** This function removes the need for the traditional Tuxedo user-security database file, `tpusr`. Instead, WebLogic Server's embedded LDAP stores all of the information required for Tuxedo user authentication. At runtime, a Tuxedo authentication server, `LAUTHSVR`, retrieves user information from the WebLogic Server's embedded LDAP and authenticates the user. If the authentication is successful then an appkey is returned to the user.

15

**[0025]** In Tuxedo, a Tuxedo user can only belong to one Tuxedo group. However, in WLS a user can be a member of several groups. To resolve this problem, in one embodiment, the authentication server will put a user in the first Tuxedo group it discovers. For example, suppose a user Sam belongs to `GRP1`, `GRP2`, and `WLSGRP`. Both `GRP1`, and `GRP2` are defined in the `tpgrp` file, and thus are Tuxedo groups. `WLSGRP` is not defined in the `tpgrp` file, and it is not mapped to Administrators and Operators groups so it is not a Tuxedo group. When Sam logs into Tuxedo, the authentication server finds Sam belongs to `GRP1` before it finds out Sam also belongs to `GRP2`; then Sam will be assigned the APPKEY with value of `GRP1`.

20

25

**[0026]** A valid Tuxedo group is a group defined in the `tpgrp` file plus the

Administrators and Operators groups. By default Administrators maps to WLS "Administrators" group, and Operators maps to WLS "Operators" group. So by default a WLS Administrators user can also administrator Tuxedo, a WLS Operators user can operate Tuxedo. This default mapping can be modified  
5 through the configuration file. By changing the mapping a Tuxedo administrator may not be a WLS administrator, and vice versa.

#### Single Security Administration Console

**[0027]** In accordance with one embodiment, the single security system  
10 and methods can be used to allows the system administrator to administer the security database from within the WLS console for Tuxedo. This features uses the single data repository for user security information as described above. The administrator only needs to configure the user and user password once in WLS instead of configuring the user in both Tuxedo and WLS. This means the  
15 administrators can configure all of their Tuxedo and WLS users from a single WLS admin console. Since the single authorization is not required, this approach only stores user name and user password information in WLS LDAP. Tuxedo continues to hold the information of group and access control list. This is simpler since it only uses WLS to resolve user authentication.

20 **[0028]** The present invention may be conveniently implemented using a conventional general purpose or a specialized digital computer or microprocessor programmed according to the teachings of the present disclosure. Appropriate software coding can readily be prepared by skilled  
25 programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art.

**[0029]** In some embodiments, the present invention includes a computer program product which is a storage medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the processes of the present invention. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical discs, DVD, CD-ROMs, microdrive, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, DRAMs, VRAMs, flash memory devices, magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data.

**[0030]** The foregoing description of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations will be apparent to the practitioner skilled in the art. Particularly, while the embodiments of the system described above are described in the context of WebLogic and Tuxedo servers, it will be evident that the system may be used with other types of applications, clients, application servers, and enterprise servers. It will also be evident that the system can be used to provide security for users and for user/client applications, and that the system may be extended to include a plurality of servers. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention for various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalence.